

**REMARKS**

Claims 1-29 are pending in the application. Claims 1-29 were rejected under 35 U.S.C. §102(a) as being anticipated by Richter et al. Claims 1, 2, 4-7, 12, 13, 15-18, 23, 24, and 26-29 have been amended. Reconsideration and reexamination of the application in view of the amendments and following remarks is respectfully requested.

Claims 1-29 were rejected under 35 U.S.C. §102(a) as being anticipated by Richter. Independent claims 1, 12 and 23 have been amended. With the amendments to these claims, it is respectfully submitted that the rejection has been overcome.

The present invention is directed to managing a shared read/write buffer pool and the execution of read and write commands to ensure that free blocks are available to temporarily store read or write data. A read command typically results in a number of pending read data requests that are satisfied over time, each of which requires the use of blocks in the shared read/write buffer pool. Without the management provided by the present invention, during the pendency of the read command, write commands may be initiated that consume the remaining buffer pool resources such that there will not be enough buffer pool resources to satisfy the pending read data requests. With both the pending read data requests and pending write commands consuming all of the buffer pool resources, neither may be able to complete, and a lockup condition may occur. To reduce the amount of read data re-transmissions, the write command may be throttled based upon the amount of pending read data requests that are currently unsatisfied and the amount of free blocks available. If the currently available free blocks would be substantially consumed by the total outstanding inbound read data requested, no more write commands will be initiated. As inbound read data is received into allocated buffers and transferred to the initiator device, the blocks in the buffer pool are freed up. When the read data transfer is completed or sufficient buffer resources have been freed up, the write data command may be initiated.

Claims 1, 12 and 23 have been amended to more particularly recite that the buffer pool is a shared read/write buffer pool. It is this shared read/write buffer pool that results in the execution of read and write commands to be dependent on each other. Furthermore, because of the special

nature of read commands (which result in a number of pending read data requests), claims 1, 12 and 23 have also been amended to more particularly recite that a new write data command is not initiated until enough pending read data requests have been processed to free up enough blocks in the shared buffer pool to accommodate the new write data command.

Richter fails to disclose managing a shared read/write buffer pool. Because Richter fails to disclose managing a shared read/write buffer pool, the special problems created by the shared buffer pool are not present in Richter, and are therefore not addressed in Richter. In particular, Richter does not discuss the possibility that multiple pending read data request resulting from a single read data command along with new write commands may consume the available blocks in the buffer pool and create a lockup condition, allowing neither operation to complete. Richter only discloses resource management in very general terms. (See FIG. 5 and paragraphs [0211] through [0234].) Buffer pool utilization is mentioned (see paragraph [0021]), but Richter does not disclose, teach or suggest that the buffer pool is shared between read and write commands. Resource management is described only at a high level in paragraphs [0211] through [0234], and nowhere is the required staggered execution of read and write commands discussed (i.e. holding off on subsequent write commands until sufficient pending read data requests have been completed).

In particular, Richter contains no disclosure at all related to “a shared read/write buffer pool of blocks for temporarily storing write data to be sent to a peer device and read data received from the peer device” as recited in claims 1, 12 and 23. Richter also is completely silent as to “preventing an initiation of a new write data command until pending read requests have been processed enough to free up sufficient blocks in the buffer pool to accommodate the data of the new write data command,” as recited in claims 1, 12 and 23.

Because Richter does not disclose all of the limitations of amended claims 1, 12 and 23, it is respectfully submitted that the rejection of claims 1, 12 and 23 under 35 U.S.C. §102(a) as being anticipated by Richter has been overcome.

In addition, because claims 2-11 depend from claim 1, claims 13-22 depend from claim 12, and claims 24-29 depend from claim 23, it is respectfully submitted that the rejection of those claims has been overcome for the same reasons provided with respect to claims 1, 12 and 23.

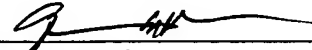
In view of the above, each of the presently pending claims in this application is believed to be in immediate condition for allowance. Accordingly, the Examiner is respectfully requested to pass this application to issue.

If, for any reason, the Examiner finds the application other than in condition for allowance, Applicants request that the Examiner contact the undersigned attorney at the Los Angeles telephone number (213) 892-5752 to discuss any steps necessary to place the application in condition for allowance.

In the unlikely event that the transmittal letter is separated from this document and the Patent Office determines that an extension and/or other relief is required, Applicants petition for any required relief including extensions of time and authorizes the Commissioner to charge the cost of such petitions and/or other fees due in connection with the filing of this document to **Deposit Account No. 03-1952** referencing Docket No. 491442001600.

Dated: May 16, 2005

Respectfully submitted,

By   
Glenn M. Kubota  
Registration No.: 44,197  
MORRISON & FOERSTER LLP  
555 West Fifth Street, Suite 3500  
Los Angeles, California 90013  
(213) 892-5200